UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
08/987,849	12/09/1997	JOHN V. MCLAIN JR.	COS-97-033	6786

25537　　7590　　04/23/2003

WORLDCOM, INC.
TECHNOLOGY LAW DEPARTMENT
1133 19TH STREET NW
WASHINGTON, DC 20036

EXAMINER
LAO, SUE X

ART UNIT	PAPER NUMBER
2126	

DATE MAILED: 04/23/2003

13

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 07-01)

	Application No.	Applicant(s)
Office Action Summary	08/987,849	McLain, Jr. et al
	Examiner	Art Unit
	S. Lao	2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE ___3___ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *Jan 23, 2003* _____ .

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-18* _____ is/are pending in the application.

   4a) Of the above, claim(s) *7-13* _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *1-6 and 14-18* _____ is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claims _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are a) ☐ accepted or b) ☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11) ☐ The proposed drawing correction filed on _____ is: a) ☐ approved b) ☐ disapproved by the Examiner.

   If approved, corrected drawings are required in reply to this Office action.

12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13) ☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a) ☐ All b) ☐ Some* c) ☐ None of:

   1. ☐ Certified copies of the priority documents have been received.

   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

   3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   *See the attached detailed Office action for a list of the certified copies not received.

14) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

   a) ☐ The translation of the foreign language provisional application has been received.

15) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s). _____

4) ☐ Interview Summary (PTO-413) Paper No(s). _____

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____

DETAILED ACTION

1.   Claims 1-18 are pending. This action is in response to the amendment and remarks filed 1/23/2003. Applicant has added claims 1-6, 14 and 18.

2.   The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3.   Upon consideration of applicant's argument that claims 15-18 are the means plus function claims of the originally filed claims 1-4, the rejections of claims 15-18 are now included in the office action. Applicant is required to cancel non-elected claims 7-13.

4.   The non-statutory double patenting rejection, whether of the obviousness-type or non-obviousness-type, is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent. *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969); *In re Vogel,* 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Van Ornum,* 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Longi,* 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); and *In re Goodman,* 29 USPQ2d 2010 (Fed. Cir. 1993).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(b) and © may be used to overcome an actual or provisional rejection based on a non-statutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.78(d).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

5.   Claims 1-6, 14-18 are rejected under the judicially created doctrine of obviousness - type double patenting as being unpatentable over claims 1-23 of U.S. Patent No.

6,295,518 to McLain et al, or over claims 1-18 of U.S. Patent No. 6,256,659 to McLain et al. Although the conflicting claims are not identical, they are not patentably distinct from each other. For example, the first and second command control vectors of the present application are met by the two or more command control vectors of U.S. Patent No. 6,295,518, the single copy of the method object is met by the command response table of U.S. Patent No. 6,295,518 as well as by the virtual object of U.S. Patent No. 6,256,659, the script is met by the script invocation of U.S. Patent No. 6,256,659 as well as by the script file database of U.S. Patent No. 6,295,518, the first/second data object are met by the data objects of U.S. Patent No. 6,295,518 as well as by the first/second data elements of U.S. Patent No. 6,256,659, the n/m logical units of instructions are met by the n/m logical units of computer instructions of U.S. Patent No. 6,256,659 as well as by the n/m logical units of work of U.S. Patent No. 6,295,518.

6.   Claim 16 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 16 recites three method step limitations (lines 5-9) in a means plus function type of claim: claim 16. For the purpose of art rejection, these method step are interpreted as corresponding means plus function limitations, as best understood and as they appear to be.

7.   Claims 1-3, 14-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burgess (U S Pat. 5,652,888) in view of Gamma et al (Design Patterns, Singleton Pattern, pages 127-134).

As to claim 15, Burgess teaches managing computer system resources (invoke target objects in a windowing environment), comprising the means' for

(1) generating a first command control vector for a first input message (message/event, e.g., LineUp, from parent window), the first command control vector identifying (713) a method object (connection manager CObject) that contains one or more

instructions for processing the first input message (send the message by invoking NotifyEvent of target object / child window) [col. 8, line 46 - col. 11, line 12];

(2) generating a second command control vector associated with a second input message (message/event, e.g., Value), the second command control vector identifying the same method object (connection manager CObject) identified by the first command control vector, the method object containing one or more instructions for processing the second input message (send the message by invoking NotifyEvent of target object / child window) [col. 8, line 46 - col. 11, line 12];

(4) processing the first and second input messages using the same method object (send the message by invoking NotifyEvent of target object / child window).

While Burgess always uses the same method object (connection manager CObject) to process different messages (col. 9, lines 1-2, 10-11), Burgess does not teach providing only a single copy of the method object.

Gamma teaches object creation, including creating a class (singleton) to provide only a single copy of the class object (class having one instance). See page 127-128. Gamma further teaches a typical application of the singleton object creational pattern is to make a window manager into a single copy/instance (page 127, section Motivation). Given the teaching of Gamma, it would have been obvious to provide a single copy of the method object (a window manager) of Burgess. The motivation to combine the teachings of Burgess and Gamma also includes the following. Burgess desires to hide different implementations / functional prototypes of the target object from the source object such that each source member function invokes target member functions with the same prototype. Col. 4, lines 31-37. Burgess does not provide a mechanism for calling with the same prototype. Gamma, on the other hand, provides a mechanism for calling a target object with the same prototype (singleton, to permit refined/various/controlled underlying operations and representations). Page 128. Therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Gamma to achieve the hiding in Burgess.

As to claims 16 and 17, Burgess teaches identifying a communication link from which the first input message is received (input port), a destination device for which the

message is intended (output port), [col. 4, lines 50-67], identifying the method object (identified member function of the target object), identifying a first current instruction used to process the input message (pointer to the target member function). See col. 4, lines 19-54; col. 7, lines 3-54. Burgess uses this logic to process all the messages / command control vectors, including the first and the second. Note the discussion of claim 15 for identifying the same method object.

As to claim 1, Burgess teaches managing computer system resources (invoke target objects in a windowing environment), comprising the steps of

(1) generating a first command control vector for a first input message (message/event, e.g., LineUp, from parent window), the first command control vector identifying (713) a method object (connection manager CObject) that contains one or more instructions for processing the first input message (send the message by invoking NotifyEvent of target object / child window) [col. 8, line 46 - col. 11, line 12];

(2) generating a second command control vector associated with a second input message (message/event, e.g., Value), the second command control vector identifying the same method object (connection manager CObject) identified by the first command control vector, the method object containing one or more instructions for processing the second input message (send the message by invoking NotifyEvent of target object / child window) [col. 8, line 46 - col. 11, line 12];

(4) processing the first and second input messages using the same method object (send the message by invoking NotifyEvent of target object / child window).

While Burgess always uses the same method object (connection manager CObject) to process different messages (col. 9, lines 1-2, 10-11), Burgess does not teach to provide only a single copy of the method object.

Burgess further teaches identifying the method object (identified target object), identifying a first current instruction used to process the first input message (identified member function of the identified target object). See col. 2, lines 33-35; col. 4, lines 19-54; col. 7, lines 3-54. It is noted that Burgess uses this same logic to process all the messages / command control vectors, including the first and the second. It is further noted that in an

object-oriented system, by passing an event object to an identified member function of an identified target object, the event is processed by the identified member function of the identified target object.

Gamma teaches object creation, including creating a class (singleton) to provide only a single copy of the class object (class having one instance). See page 127-128. Gamma further teaches a typical application of the singleton object creational pattern is to make a window manager into a single copy/instance (page 127, section Motivation). Given the teaching of Gamma, it would have been obvious to provide a single copy of the method object (a window manager) of Burgess. The motivation to combine the teachings of Burgess and Gamma also includes the following. Burgess desires to hide different implementations / functional prototypes of the target object from the source object such that each source member function invokes target member functions with the same prototype. Col. 4, lines 31-37. Burgess does not provide a mechanism for calling with the same prototype. Gamma, on the other hand, provides a mechanism for calling a target object with the same prototype (singleton, to permit refined/various/controlled underlying operations and representations). Page 128. Therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Gamma to achieve the hiding in Burgess.

As to claims 2 and 3, Burgess teaches identifying a communication link from which the first input message is received (input port), a destination device for which the message is intended (output port), [col. 4, lines 50-67], identifying the method object (identified member function of the target object), identifying a first current instruction used to process the input message (pointer to the target member function). See col. 4, lines 19-54; col. 7, lines 3-54. Burgess uses this logic to process all the messages / command control vectors, including the first and the second. Note the discussion of claim 1 for identifying the same method object.

As to claim 14, it is a program product claim of claim 1. Note claim 1 for discussion.

8.   Claims 4, 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burgess in view of Gamma et al as applied to claims 1, 15 and further in view of author admitted prior art APA (page 2, line 25 - page 3, line 20)

As to claims 4, 18, APA teaches a method object invoking a script (page 2, line 29). Given the teaching of APA, it would have been obvious to implement a method object of Burgess with a script. Note discussion of claim 1 for a single copy. The combined teachings would have provided a single copy of a script. The motivations to combine the teachings includes the following. Burgess teaches GUI objects which are typically used in a morden windowing environment. To one of ordinary skill in the art, a morden windowing environment is often a multi-tasking system which requires interrupt mechanisms to provide the preemption. Burgess does not provide such a mechanism. APA, on the other hand, provides an interrupt mechanism (timer interrupt) to provide multi-tasking. Therefore, one of ordinary skill in the art would have been motivated to use the interrupt mechanism of APA to achieve multi-tasking in Burgess.

9.   Claims 5-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burgess in view of Gamma et al and APA as applied to claim 4 and further in view of Carr et al ("Compiling Distributed C++").

As to claim 5, Burgess as modified teaches (APA) identifying current script instructions in the first and second command control vectors for processing the first and second input messages, respectively (pointer to identify next instruction to execute within a method object) (APA, page 3, lines 9-14). Note discussion of claim 4 for the method object invoking a script. Burgess as modified does not teach steps (ii) and (iii).

Carr teaches using a data object (value object) to store data generated during execution (return values) of a C++ program (pages 499-500, section 6). Given the teaching of Carr, it would have been obvious to use first/second data objects to store data generated during execution of first/second command control vectors in the C++ implementation of Burgess. It would have been obvious to combine the teachings of Burgess as modified by Gamma and Carr because the former implements the teaching in

language C++ (col. 3, lines 16-19) and the latter details one version of the language C++ (section 1).

As to claim 6, APA teaches processing a number n of logical units of instructions for a first type (process one type of instructions); interrupting such processing (timer interrupt); and processing a number m of logical units of instructions for a second type. See page 3, lines 3-8. Given the teaching of APA, it would have been obvious to alternate the processing of the first message / first command control vector and the processing of the second message / second command control vector. Note discussion of claim 4 for a motivation to combine.

10.   Applicant's arguments filed 1/23/2003 have been fully considered but they are not persuasive.

For each argued limitation (pages 9-15), applicant listed the claim limitation and the portion(s) of the prior art relied on in the rejection and simply concluded that the claimed limitation is not met by / equivalent to the prior art. Without providing specific underlying analysis as to why the portion(s) of the prior art relied on in the rejection do not support the examiner's position, this is disagreement without reason.

Regarding identifying the method object and identifying a first current instruction used to process the first input message as recited in claim 1, applicant argued that is not taught by Burgess because Burgess uses a pointer to identify the target member function (pages 9-10). The examiner respectfully disagrees. Firstly, the language of claim 1 does not provide any details of identifying except for reciting identifying. See claim 1, lines 6-9. Therefore, the language of claim 1 does not preclude the identifying with a pointer. Secondly, Burgess teaches identifying the method object (identified target object) and identifying a first current instruction used to process the first input message (identified member function of the identified target object). See col. 2, lines 33-35; col. 4, lines 19-54; col. 7, lines 3-54. It is noted that in an object-oriented system such as Burgess', by passing an event object to an identified member function of an identified target object, the event is processed by the identified member function of the identified target object. Thirdly,

throughout the application as filed, applicant discloses that the present application uses a pointer to identify the object and the instruction. See, for example, page 4, lines 12-15, 20-25; "Field 916 contains a pointer to a method object... Field 918 contains a pointer to a particular instruction within the method object ..." (Page 19, lines 18-21). In other words, Burgess meets the identifying as claimed as well as disclosed.

As to the argument that Burgess does not teach the second command control vector and second current instruction and the office action merely states that Burgess "uses this same logic to process all the messages / command control vectors, including the first and the second", it is respectfully pointed out that in the discussion of claim 1, the office action provided specific details of how the second command control vector and second current instruction are met by Burgess: "(2) generating a second command control vector associated with a second input message (message/event, e.g., Value), the second command control vector identifying the same method object (connection manager CObject) identified by the first command control vector, the method object containing one or more instructions for processing the second input message (send the message by invoking NotifyEvent of target object / child window) [col. 8, line 46 - col. 11, line 12];" (see previous office action, page 4, second paragraph).

Regarding the motivation to combine the teachings of Burgess and Gamma (page 11), a further explanation is provided as "The motivation to combine the teachings of Burgess and Gamma also includes the following. Burgess desires to hide different implementations / functional prototypes of the target object from the source object such that each source member function invokes target member functions with the same prototype. Col. 4, lines 31-37. Burgess does not provide a mechanism for calling with the same prototype. Gamma, on the other hand, provides a mechanism for calling a target object with the same prototype (singleton, to permit refined/various/controlled underlying operations and representations). Page 128. Therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Gamma to achieve the hiding in Burgess.".

Regarding the argued communication links and destination (page 12), these are met by the input and output ports identified by the respective pointers. See Burgess, col. 4, lines 50-67.

Regarding the argued features of the admitted prior art relied on, ie, script and m/n instructions (pages 13, 15), it is the combined teaching including the admitted prior art, instead of the prior art alone, that meets the claimed script and m/n instructions. See discussions of claims 4 and 6 for detail.

Regarding Carr, applicant argued that Carr does not teach data objects as claimed (page 14) and the motivation to combine. The examiner's response is that it is the combined teachings of Burgess, Gamma and Carr, instead of Carr alone, that meets the claimed data objects.

11.   The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. "An Object-Oriented Approach to Software Design for Distributed Real-time Computing Systems" by S. Yau teaches using a single method called virtual method to encapsulating multiple actual methods, thus providing a virtual object for processing multiple input messages.

12.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The examiner's supervisor, SPE Alvin Oberley, can be reached on (703) 305 9716. The examiner can normally be reached on Monday - Friday, from 9AM to 5PM. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7238 for After Final communications, (703) 746-7239 for Official communications and (703) 746-7240 for Non-Official/Draft communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9600.

Sue Lao   *Sue Lao*
April 7, 2003